Its All Graph To Me: Foundational Topology Models with Contrastive Learning on Multiple Domains

Alex O. Davies [†], **Riku Green** [†], **Nirav S. Ajmeri** [†], **and Telmo M. Silva Filho** [‡]
[†]School of Computer Science
[‡]School of Engineering Mathematics and Technology
University of Bristol, UK
{alexander.davies,zs18656,nirav.ajmeri,telmo.silvafilho}@bristol.ac.uk

Abstract

Representations and embeddings of graph data have been essential in many domains of research. The principal benefit of learning such representations is that the pre-trained model can be fine-tuned on smaller datasets where data or labels are scarce. Existing models, however, are domain specific; for example a model trained on molecular graphs is fine-tuned on other molecular graphs. This means that in many application cases the choice of pre-trained model can be arbitrary, and novel domains may lack an appropriate pre-trained model. This is of particular issue where data is scarce, precluding traditional supervised methods.

In this work we use adversarial contrastive learning to present FoToM, a model pre-trained on many graph domains. We train the model only on topologies but include node labels in evaluation. We evaluate the efficacy of its learnt representations on various downstream tasks. Against baseline models pre-trained on single domains, as well as un-trained models and non-transferred models, we show that performance is equal or better using our single model. This includes when node labels are used in evaluation, where performance is consistently superior to single-domain or non-pre-trained models.

1 Introduction

Graphs, as a general form of relational data, are ubiquitous in almost all domains. Graphs themselves are a challenging form of data to approach, as methods normally applied to continuous or categorical data-types aren't applicable, and the high dimensional nature of graphs makes sufficiently expressive models difficult to develop. As such much work has been conducted on forming useful representations of graphs which are more easily used by well explored methods. Khoshraftar et al. [18], Hamilton et al. [12] and Chen et al. [4] provide surveys on graph representation learning in general. Cai et al. [3] do the same for graph embedding (ie specifically vector representations). Besta et al. [1] provide a survey on lossless graph compression and space-efficient representations. Alternatively, many works are domain-specific, with for example Jiao et al. [15] reviewing graph representations intersection with computer vision, Li et al. [21] doing the same for biomedicine and healthcare, and a more specific example with Fasoulis et al. [9] providing a review of graph representation for proteomics.

Notably, to the best of our knowledge, there have been no models proposed that learn representations for multiple domains of graphs. The current state-of-affairs is instead domain-specific. In this work we present FoToM (**Fo**undational **To**pology **M**odel), a single-model capable of representing multiple domains of graphs simultaneously, trained through adversarial graph constrastive learning [29]. FoToM is a foundational graph model which can be fine-tuned on smaller datasets with domain-specific node labels. FoToM is available as a PyPI package[1].

[1]https://pypi.org/project/fotom/

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).

Node labels are excluded during training to ensure that FOTOM learns generalised graph structures. During evaluation we include node labels, and demonstrate that the multi-domain FOTOM model out-performs both non-pretrained models and domain-specific models trained in the same manner. Further, in an intriguing finding, we find that it out-performs the aforementioned models when node labels are included under fine-tuning. This demonstrates that the updates and aggregations learnt by FOTOM on generic topologies are easily transferred to tasks where node labels are essential features.

2 Related Work

Representation learning, in most cases, aims to compress data into a numeric vector $\vec{z}$. For non-tabular data, for example images [30] and text [17], this allows more traditional statistical or ML methods to be applied. This includes, as an example, semantic textual similarity on natural language data. On images and text many approaches have been researched for forming useful representations. Earlier approaches broadly consisted of supervised training on a given task, then using a hidden layer's latent space as representations. More recently the inherent order of such data, pixel-to-pixel or word-to-word, allows fairly intuitive approaches such as "predict the missing chunk" tasks for representation learning. Graph structures lack such inherent ordering, somewhat precluding missing-chunk style pre-training tasks. As such unsupervised learning, with no downstream predictive task in the training process, has become the de-facto area of reasearch.

Hamilton's work provides a comprehensive summary of methods for Graph Representation Learning (GRL) [11]. Traditional methods to encode graphs as vectors include graph statistics such as centrality measures and clustering coefficients, but also kernel methods such as the Weisfeiler-Lehman (WL) kernel [26]. Graph Neural Networks (GNNs) were introduced to address the limitations of traditional methods to encode graphs by using the deep learning mechanisms to 'self-learn' important features for a given graph. GNNs use a message-passing mechanism, much like how the WL-kernel operates. GNNs are shown to be, at most, equally expressive. However they are differentiable, can handle continuous node features, and are better suited for inductive tasks, making GNNs the current state-of-the-art for learning to generalise to new or unseen graphs. The WL kernel is still relevant and is used to test for isomorphism, by evaluating how many iterations of the WL-algorithm is used before two graphs contain differing representations [11].

Unsupervised representation learning, as a pre-trained model, has been shown to be effective for image data, but for graph data the results are less clear. Randomly initialised (untrained) GNNs, which are used as graph-kernels, have been shown in some specific cases to be just as effective as a pre-trained graph encoder [31]. Untrained neural networks have similarly been shown to be effective encoders for many data structures, with careful encoder design essential to form more useful representations.

2.1 Contrastive Learning

Contrastive learning aims to learn useful representations of input samples without the necessary consideration of downstream tasks. We denote a set of real data $X = \{x_i, x_j, x_k, \ldots\}$. The most common formulation, as proposed by Chen et al. [5], is fairly simple. For each input sample, duplicate it and apply different augmentations to both, $\tilde{X} = \{x'_i, x''_i, \ldots\}$, keeping track of this 'positive pair' both derived from the same input sample. In the broadest formulations a pair of samples $\{x'_i, x'_j\}$ where $i \neq j$ is a 'negative pair'. The encoder produces an embedding for each, denoted $\tilde{H} = \hat{H}(\tilde{X}) = \{h'_i, h''_i, \ldots\}$, then then cosine similarity between them is $L_i = (h'_i \cdot h''_i)/(|h'_i| \cdot |h''_i|)$. This is, in its simplest form, the loss function being optimised. Chen et al. [5] achieve SOTA performance with minimal training data using representations from images with this approach after fine-tuning.

2.1.1 Graph Contrastive Learning

Multiple studies exist for graph contrastive learning [6, 10, 13, 14, 20, 35, 37]. These only consider applying transfer learning for downstream tasks on the same domain of data, for example pre-training and transfer only on molecular graphs. To the best of the authors' knowledge, there have not been any previous studies exploring pre-trained multi-domain graph embeddings. Previous authors have argued, but not shown, that an ImageNet equivalent for graphs is not feasible [27].

Graph contrastive learning (GCL) takes the following stages: encoder-function selection, augmentation strategy, negative-sample selection, and contrastive learning loss [37, 35, 34]. Comparisons of different GNN architectures as encoders find that the Graph Isomorphism Network (GIN) consistently produces the most useful representations [14, 37]. It has been shown that the quality of augmentations is critical for GCL, and simple Bernoulli-based augmentations do not produce the most effective representations [37, 35]. Many works have addressed improving Bernoulli-based augmentations by considering graph properties [36, 28, 38, 29]. Zhang et al. [36] and Subramonian [28] consider important motifs for chemical-data graphs and find that dropping non-essential edges improves representation learning.

Many of these works find that having fixed-parameter augmentations, for example a set random probability of dropping each edge, leads to highly volatile performance as said parameter varies. As an example, taking a molecule, dropping a single edge can fundamentally alter chemical properties. This leads to a very narrow band of edge-dropping probabilities that yield good results. Too high a probability and the resulting augmented molecule can be altered to be drastically different to the source molecule, and too low and the augmentations are effectively impotent. On our multi-domain task this is even more of an issue, as graphs should cover greatly varied sizes and densities, so a single fixed parameter is highly unlikely to suffice.

2.1.2 Adversarial Graph Contrastive Learning

An intuitive and potentially more flexible idea we use here is that of *learnt* augmentations, specifically as proposed by Suresh et al. [29]. Their ADversarial-Graph Contrastive Learning (AD-GCL) method trains both an encoder and a 'view-learner', the latter of which learns to drop edges selectively, aiming to reduce mutual information between the augmented graph and its original source graph.

AD-GCL comprises of two models: an encoder f_Θ and a view-learner $T_\Phi(G)$ where Θ, Φ represent learnt parameters. The first term of their objective function minimizes the mutual information between the encoding of the true graph and its augmentation from the view learner. The second term λ_{reg} regularises the view learner, penalising linearly with the proportion of edges dropped during augmentation. This encourages meaningful augmentations through the trade-off between mutual information and the proportion of dropped edges. The ability to learn augmentations ("views") specific to each domain simultaneously is vital for FOTOM.

3 Method

This work focuses only on labelled graph topologies, meaning a graph set is composed of a node-set $V : \{v_1, v_2, ...\}$ and the (un-weighted, un-directed) edges between those nodes $E : \{(v_i, v_j), (v_i, v_k), ...\}$. As an effort to aid downstream transfer learning, we include the same "dummy" node label on all nodes. Using the integer 1 here is essentially identical to passing no node labels. These dummy labels can then be replaced with accurate node labels (eg. atom types) during fine-tuning. As such the model we aim to produce should learn from generic and hopefully complex graph topological structures. Such structures represent a significant portion of the information that any GNN model must learn to represent, regardless of domain.

As such fine tuning FOTOM, instead of using a bespoke model for each domain, should alleviate computational costs and potentially bolster performance in domains where data is scarce. During early experimentation we included node labels from each dataset in the training data, but found that as said numeric labels represent different information in each dataset, the model simply learnt from node label distributions instead of graph characteristics.

3.1 Dataset

The aim of this work is to produce a model that has learnt useful representations of graphs *in general*, not on a specific domain. We achieve this by constructing a dataset that as broadly as feasible covers real-world graphs, although here we consider only topologies (ie. no node, graph or edge features, including labels). Features beyond node labels are left as an area for future research. Constructing a dataset of real-world graphs is itself a reasonable narrowing of scope, but as the possible space of graphs at each number of nodes is so massive, constraining ourselves to a large and highly varied set of graphs is a reasonable step. Such a dataset should contain graphs that present a large diversity of

Table 1: Statistics for each of our datasets. If used for training, we denote these statistics averaged across the training data. Otherwise we report statistics for their validation datasets.

	Num. Graphs	Num. Nodes	Num. Edges	Diameter	Avg. Clustering
Training Only					
molpcba	250000	25.7 ± 6.28	27.7 ± 7	13.5 ± 3.29	0.00128 ± 0.0115
Training/Evaluation					
Twitch	50000	29.6 ± 11.1	86.6 ± 70.7	2 ± 0	0.549 ± 0.149
Facebook	50000	59.5 ± 20.7	206 ± 170	10.2 ± 6.39	0.429 ± 0.133
Cora	50000	59.3 ± 20.7	119 ± 56.6	10.1 ± 4.68	0.322 ± 0.0845
Roads	50000	59.5 ± 20.8	73.9 ± 28.2	16.4 ± 5.92	0.0559 ± 0.0426
Fly Brain	50000	59.5 ± 20.8	146 ± 99.6	8.8 ± 3.49	0.237 ± 0.0875
Evaluation Only					
molesol	113	17.7 ± 6.4	19.4 ± 7.24	8.24 ± 3.1	0.00136 ± 0.0101
molclintox	148	32.7 ± 18.7	35.6 ± 19.6	14.3 ± 6.81	0.00522 ± 0.0187
molfreesolv	64	12 ± 4.45	12.7 ± 5.15	5.98 ± 2.01	0 ± 0
mollipo	420	27.3 ± 8.48	30 ± 9.24	13.9 ± 4.16	0.00469 ± 0.021
Trees	5000	19.6 ± 6.96	18.6 ± 6.96	10.1 ± 3.11	0 ± 0
Random	5000	29.3 ± 10.7	111 ± 82.1	3.88 ± 1.16	0.227 ± 0.0998
Community	5000	48 ± 0	323 ± 26.2	3 ± 0.0346	0.408 ± 0.0256

multi-node patterns. Our approach is simple: include graphs from many domains, and assume that there is adequate coverage of the multi-node patterns that occur in real-world graphs. This assumption is evaluated through unseen-domain downstream tasks. The same approach has essentially been taken in the domains of text [2] and images [23], where enormous numbers of data samples are scraped from the web for generative pre-training, with the implicit assumptions that **a)** this dataset adequately covers the relevant space, and **b)** a single model can learn such a broad space.

The datasets we use are described in aggregate measurements in Table 1. **molpcba [14]** A dataset of 437,929 small molecules selected from MoleculeNet [33]. **molesol, molclintox, molfreesolv, mollipo [14]** Smaller datasets sampled from MoleculeNet, with single-target classification or regression downstream tasks. **Facebook [24]** A single graph of page-page connections on Facebook. Originally 22,470 nodes and 171,002 edges. Downstream task set to be predicting average clustering. **Twitch [25]** Ego networks from the streaming platform Twitch. Ego networks have one central "hub" node, which shares an edge with all other nodes. This dataset has a downstream binary classification task. **Cora [22]** A much used citation graph. 2,810 nodes and 7,981 edges. Downstream task set to be predicting the number of 4-cycles in the graph normalised by graph size. **Roads [19]** The road network of Pennsylvania. Junctions are nodes, of which there are 1,088,092, with edges roads between them, of which there are 1,541,898. Downstream task set to be predicting graph diameter normalised by graph size. **Fly Brain [32]** The full neural connectome of a fruit fly larvae. Each node is a neuron, of which there are 3,016. Each edge is a synapse between neurons, but as the multi-graph is dense at originally 548,000 edges, we only include an edge between neurons if there are more than two synapses between neurons. Downstream task set to be predicting the number of 5-cycles in the graph normalised by graph size. **Trees** Trees, of a randomly sampled maximum depth, and a randomly fixed probability of branching at each level. Downstream tasks is predicting depth. **Random** Random Erdos-Renyi (ER) graphs [8], with a number of nodes and edge probability randomly sampled. These could represent noise during training, but predicting edge probabilities from embeddings represents a useful bare-minimum downstream task during validation and testing. **Community** Small ER subgraphs, with a set intra-subgraph edge probability, and a random inter-subgraph connection probability. Like our Random dataset, this is employed for downstream validation with the inter-subgraph connectivity as target.

As the Facebook, Cora, Roads and Fly Brain datasets consist of a single graph, we employ Exploration Sampling With Replacement [7] to construct datasets of many smaller graphs. We use a randomly

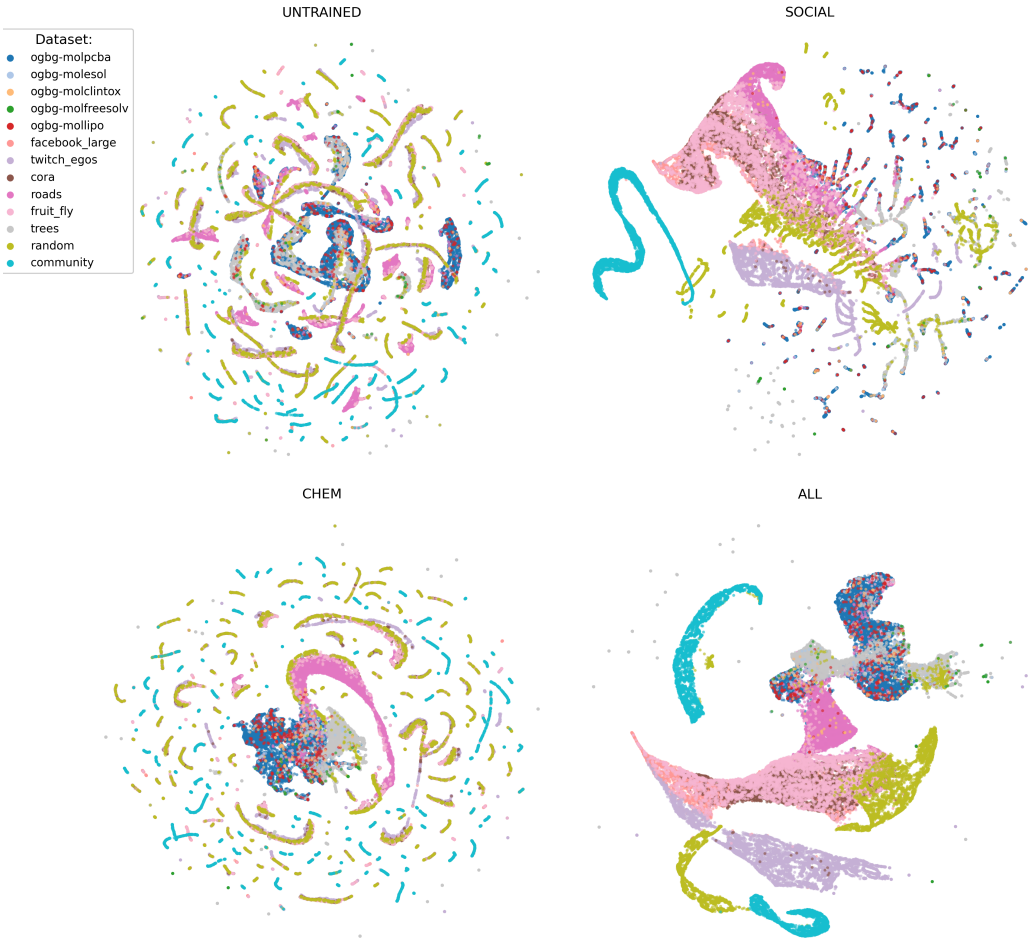


Figure 1: A UMAP embedding of encodings from each model, as well as an untrained model. The untrained and Chem models are noticeably more fragmented than the Social and All models. In turn the Social model is more fragmented than the All model. In the All model embedding, molecules are all in the same region, furthest from the Twitch Ego networks, with the Cora and Facebook Page-Page data "filling the gap".

selected exploration sampler[2] for each sample, which in turn samples a random number of nodes in the range $24 \leq |V| \leq 96$ from the source graph. The use of a selection of exploration samplers should further ensure that a large variety of graphs and topological features are included in the datasets.

Taking downstream task on each validation set where available, we simply sum their relevant scores (MSE and $1-$ AUROC) as a crude monitor for the model's representational abilities. Taking 50,000 samples for each training set and 5,000 for the validation sets, we conduct a limited Bayesian hyperparameter sweep. The heuristic used is a simple summing of evaluation scores on each dataset weighted by the number of samples in that dataset. Taking these hyper-parameters we fit an AD-GCL model on the large dataset ('All', i.e. all domains), a second only on molecules ('Chem'), and a third only on non-molecular graphs ('Social'). We train each model for 100 epochs with a batch size of 512. We present a UMAP embedding of encodings of the validation datasets for an untrained model, and the trained models, in Figure 1. Figure 6 shows the same for PCA projections in the Appendix.

On the combined encodings of the validation sets we compute R^2 correlation coefficients between the first five principal components (PCA) and common graph-level metrics. High correlations, or the

[2]Metropolis Hasting Random Walk, Diffusion and Forest Fire samplers, from `https://little-ball-of-fur.readthedocs.io/en/latest/`

lack thereof, should give some indication of how embedding components represent different graph characteristics. The metrics we use are as follows: **Num. Nodes** the number of nodes in a graph, **Num. Edges** the number of edges in a graph, **Density** the proportion of possible edges that exist, **Diameter** the diameter across the largest connected component of the graph, **Avg. Degree** average edges per node, **Avg. Clust.** the average proportion of triangles for a node that are complete. The results are presented, along with each component's explained variance, in Table 2, and visualised in the Supplemental material in Figure 3. The explained variance ratio is the proportion of the original data's total variance expressed by the PCA component.

4 Experiments

Table 2: PCA components, fit on the embeddings of the test set, and their most correlated graph-level metric, for each of the trained models. **Underlined** text indicates a strong correlation $|R^2| \geq 0.66$, and **bold** text indicates moderate correlation $0.33 \leq |R^2| \leq 0.66$. We show up to the first five PCA components, but include only three where explained variance values are minimal.

	Variance Ratio	Metric Correlations					
		Most	R^2	Second	R^2	Third	R^2
Untrained							
PCA 0	9.96×10^{-01}	Num. Edges	0.219	Trans.	0.072	Density	0.063
PCA 1	3.52×10^{-03}	Num. Edges	$\mathbf{-0.555}$	Trans.	-0.276	Density	-0.243
PCA 2	1.13×10^{-04}	Num. Edges	**0.390**	Trans.	0.224	Density	0.210
Chem							
PCA 0	9.97×10^{-01}	Num. Edges	0.223	Trans.	0.074	Density	0.065
PCA 1	2.60×10^{-03}	Num. Edges	$\mathbf{-0.514}$	Trans.	-0.253	Density	-0.225
PCA 2	1.14×10^{-05}	Num. Edges	0.132	Num. Nodes	0.051	Trans.	0.047
Social							
PCA 0	9.99×10^{-01}	Num. Edges	0.127	Trans.	0.032	Density	0.032
PCA 1	9.92×10^{-04}	Num. Nodes	**1.00**	Num. Edges	**0.528**	Density	$\mathbf{-0.393}$
PCA 2	4.79×10^{-08}	Num. Edges	0.133	Density	0.070	Trans.	0.054
All							
PCA 0	7.04×10^{-01}	Num. Nodes	**0.999**	Num. Edges	**0.534**	Density	$\mathbf{-0.388}$
PCA 1	1.94×10^{-01}	Num. Edges	**0.805**	Density	**0.737**	Diameter	$\mathbf{-0.554}$
PCA 2	4.23×10^{-02}	Avg. Clust.	0.309	Trans.	0.248	Num. Edges	0.235
PCA 3	2.43×10^{-02}	Diameter	**0.500**	Avg. Clust.	$\mathbf{-0.416}$	Transitivity	-0.243
PCA 4	1.82×10^{-02}	Density	-0.132	Trans.	-0.108	Diameter	-0.088

4.1 Transfer

Of course the most valid measurement of FoToM will be how well it transfers to downstream tasks. Graph level tasks, including during validation during training, can be conducted using the embeddings z that during training are passed to the projection head g.

As previously mentioned during pre-training we replace all node labels with a dummy value (specifically 1). In our evaluation of transfer learning, for each model and transfer process, we present results for both with and without node labels in the fine-tuning data. We include a simple diagram in Figure 5. This should indicate whether the transformations learnt by the encoder can be quickly re-tooled for applications where node labels contain essential data. The assumption here would be that FoToM has learnt graph structures in-general, and that under transfer the further complexity of node labels within this structure can be learnt more quickly.

4.1.1 Linear Transfer

The most simple transfer application is to take the set of encodings $\mathbb{Z} : \{z_1, z_2, ...\}$ and fit linear models for the downstream tasks. For model fitting we take 95% splits of the validation dataset in

Table 3: Scores for linear models applied to the encodings of each validation dataset, with and without node labels. For the classification datasets (ogbg-molclintox and Twitch Egos) we report AUROC, other datasets report RMSE. Some datasets, where results are very similar for each model, are excluded. **Bold** text indicates a superior result. For clarity the results for a given dataset are scaled by a constant factor.

	Untrained	**Chem**	**Social**	**All**	Magnitude
molfreesolv	4.21 ± 4.13	3130 ± 5990	2.00 ± 0.869	$\mathbf{1.01 \pm 1.02}$	10^3
molesol	48.3 ± 90.2	817 ± 1150	48.1 ± 62.9	$\mathbf{28.1 \pm 38.1}$	10^0
mollipo	29.4 ± 17.9	14.9 ± 2.0	$\mathbf{4.45 \pm 0.62}$	5.33 ± 0.56	10^0
molclintox	0.487 ± 0.005	0.440 ± 0.007	$\mathbf{0.500 \pm 0.000}$	$\mathbf{0.500 \pm 0.000}$	10^0
Facebook	10.6 ± 1.7	10.7 ± 20	7.73 ± 1.87	$\mathbf{4.44 \pm 5}$	10^{-3}
Twitch Egos	0.638 ± 0.012	0.610 ± 0.072	0.679 ± 0.002	$\mathbf{0.694 \pm 0.002}$	10^0
Roads	263 ± 0.5	256 ± 0.0	298 ± 0.3	$\mathbf{2.85 \pm 5}$	10^{-5}
Trees	7.66 ± 0.16	2.85 ± 0.0	2.98 ± 0.03	$\mathbf{2.56 \pm 0.05}$	10^{-3}
Community	4.36 ± 0.05	$\mathbf{4.25 \pm 0.00}$	12.0 ± 2	4.46 ± 0.07	10^{-5}
Random	6.46 ± 0.7	3.27 ± 0.0	$\mathbf{1.20 \pm 0.2}$	3.38 ± 0.0	10^{-4}
	Untrained-Labels	**Chem**-Labels	**Social**-Labels	**All**-Labels	
molfreesolv	$\mathbf{480 \pm 172}$	$(1.13 \pm 0.64) \times 10^6$	90900 ± 83300	11000 ± 5600	10^0
molesol	29.4 ± 7.8	$\mathbf{18.5 \pm 7.3}$	531 ± 292	34.1 ± 33.0	10^0
mollipo	7.33 ± 1.37	7.25 ± 1.01	$\mathbf{4.02 \pm 0.92}$	5.01 ± 0.53	10^0
molclintox	0.511 ± 0.022	0.497 ± 0.001	0.428 ± 0.001	$\mathbf{0.526 \pm 0.060}$	10^0
Facebook	14.6 ± 11.1	10.8 ± 1.1	7.82 ± 1.9	$\mathbf{4.43 \pm 6}$	10^{-3}

question, fit the linear model on the encodings produced by that subset, then evaluate on the whole corresponding test-set. This allows us to produce error bounds. The results are shown in Table 3.

Here we see that on the majority of datasets the model trained on the whole dataset ("All") out-performs the others. The most significant out-performance is on the Roads dataset, where the All model performs orders of magnitude better than the other models. Interestingly the untrained GIN (Graph Isomorphism Network) significantly out-performs the other models on the molfreesolv dataset, but only when node labels (here atom types) are included.

4.1.2 Full Transfer

A more complex transfer process, and in practise more likely than simply using linear models, is fine-tuning on downstream tasks and datasets. To this end we replace the projection head with a simple two-layer MLP, with a single output node. We then perform 10 fine-tuning runs on each validation dataset, with and without node labels, for each pre-trained model. The results of this transfer and fine-tuning are presented in Table 4, along with the corresponding results for the same GIN architecture without pre-training.

The All model consistently out-performs the other models here on the majority of datasets. Most significantly it does so on datasets that were shown under linear transfer (Section 4.1.1) to require specific node-label representation, such as the mol* molecular datasets. This could indicate that the representations learnt on generic, un-labelled topologies can usefully and quickly be transferred to label-specific representations.

The standard deviation of the results on each dataset are far lower than those for the non-pretrained model and the domain-specific pre-trained models. In Figures-(7a, 7b) we show validation loss on the Facebook and Twitch datasets during fine-tuning for the All model and an un-pre-trained GIN with the same architecture. Here the advantages of pre-training are clear: the pre-trained model has consistently lower validation loss, seems to plateau later and at a lower loss, and has a far lower deviation in that loss than the un-pre-trained GIN.

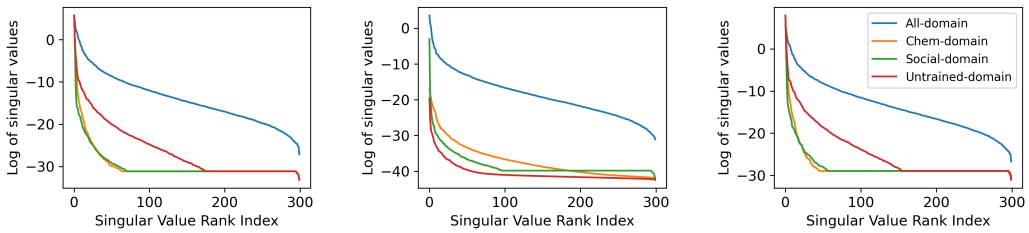
Table 4: Scores for fine-tuning the complete model on each validation dataset, with (**Model**-labels) and without (**Model**) node labels. For the classification datasets (ogbg-molclintox and Twitch Egos) we report AUROC, other datasets report RMSE. Some datasets, where results are very similar for each model, are excluded. Bold text indicates a superior result, including if error bounds overlap with another model.

	GIN	**Chem**	**Social**	**All**
molfreesolv	4.78 ± 0.711	5.12 ± 0.591	**4.08 ± 0.042**	4.22 ± 0.022
molesol	1.96 ± 0.156	1.76 ± 0.123	1.86 ± 0.08	**1.46 ± 0.062**
mollipo	1.62 ± 0.322	1.58 ± 0.081	**1.10 ± 0.008**	1.13 ± 0.012
molclintox	**0.527 ± 0.024**	0.497 ± 0.002	0.433 ± 0.010	0.468 ± 0.051
Facebook	0.311 ± 0.065	1.17 ± 0.165	**0.133 ± 0.003**	**0.134 ± 0.002**
Twitch Egos	0.72 ± 0.006	0.689 ± 0.021	0.736 ± 0.005	**0.79 ± 0.006**
Roads	0.46 ± 0.073	0.571 ± 0.065	**0.104 ± 0.001**	**0.103 ± 0.000**
Trees	0.203 ± 0.012	0.242 ± 0.025	**0.115 ± 0.000**	0.117 ± 0.001
Community	0.743 ± 0.106	0.715 ± 0.130	**0.016 ± 0.001**	**0.017 ± 0.000**
Random	0.529 ± 0.228	0.589 ± 0.089	**0.044 ± 0.000**	**0.044 ± 0.000**
	GIN-Labels	**Chem**-Labels	**Social**-Labels	**All**-Labels
molfreesolv	**3.71 ± 0.630**	5.14 ± 0.568	**4.00 ± 0.203**	**4.14 ± 0.155**
molesol	1.62 ± 0.191	1.80 ± 0.085	1.63 ± 0.146	**1.27 ± 0.107**
mollipo	1.72 ± 0.311	1.59 ± 0.084	**1.10 ± 0.016**	**1.11 ± 0.011**
molclintox	0.511 ± 0.022	0.497 ± 0.001	0.428 ± 0.001	**0.526 ± 0.060**
Facebook	0.749 ± 0.250	1.19 ± 0.250	0.140 ± 0.008	**0.134 ± 0.001**

5 Discussion

Here we discuss first the efficacy of the FoToM model presented here as a representation learner without focusing on any downstream tasks. We then present and discuss the performance of, and reasonable use cases for, the pre-trained FoToM model. Finally we detail explicitly any significant assumptions we have made and the potential risks to the validity of this work.

5.1 Dimensionality Collapse



(a) All Data (b) molpcba (c) Facebook

Figure 2: Logged singular values of the representation covariance space of pretrained models (as from Jing et al. [16] on all data, molpcba, and unseen Facebook data.

We make a task-agnostic evaluation of the learnt representation by analysing the dimensionality collapse as in [16]. As shown in Figure 7, FoToM has far fewer collapsed dimensions in comparison to single-domain pre-training and untrained encoders. This is shown across all data in Figure 2a, molecular data as in Figure 2b, and for social network data as in Figure 2c. This finding highlights that features learnt by FoToM have greater dimensional coverage than pre-training on single domains; to the best of our knowledge, this is the first record of an analysis in measuring the coverage for multi-domain pre-trained graph representation. It is possible that the increase in coverage has a relationship to the increase in performance as in Table 4, but this is a direction for future work.

5.2 Performance & Use-Cases

We have shown that pre-training on multiple domains of topologies offers significant performance improvements on downstream tasks under fine-tuning. In particular our results show that exposure to a larger variety of graph structures actually benefits performance. In Table 4, assuming that the Chem model has the least varied training data, then the Social model, then finally the All model, we can see that performance increases along the same order.

This is significant, as the most intuitive assumption would be that exposing a molecular representation learner to non-molecular graphs, which have possess very different multi-node patterns, would hinder its performance. In other words we'd normally expect that this out-of-domain data would act as noise. Instead the Social and All models out-perform the Chem model on every molecular dataset under fine-tuning. The next natural assumption would be that by omitting node labels during training the representations for molecules would lack essential information, hence the performance discrepancy we see on these un-labelled validation sets. However this line of reasoning would conclude that including node labels during fine-tuning (or supervised training of a non-pretrained model) would likely lead to these domain-specific models out-performing all-domain models. Our results instead show that performance is both superior and more consistent from our all-domain model on labelled molecular graphs, with performance increasing compared to the same model on non-labelled graphs.

As such the reasonable use-cases for this pre-trained model are broad. This potential benefit is two-fold. Firstly, pre-training and fine-tuning carries the normal decrease in expected training time. Secondly, as shown by our results in this work, our pre-trained models offer actual performance increase over simple supervised training. Our assumption is that the broad range of generic topological features learnt by the FOTOM model provide a more robust foundation for training than simple random initialisation, and hence the pre-trained models avoid overfitting to non-useful features better than randomly initialised models. Guarantees and proofs on this point are left as an area for future research.

5.3 Assumptions, Risks, Future Directions

The most significant assumption we made - that a model can learn representations that are useful on multiple domains of downstream tasks - has been justified by our results above. However, in the full scope of downstream tasks for graph data, this first exploratory work is still fairly limited. Firstly we do not experiment with different graph augmentations (or "views"), instead employing only edge dropping. Secondly our experiments are limited to a few set compositions of domains, in training a 50/50 split between molecular and non-molecular graphs for the full FOTOM model. The non-task-specific evaluation of representation models is a challenging and open research problem. The same challenge is compounded on the large and high-dimensional graph space, which in comparison to other data structures both lacks intuitive human understanding and has a far smaller body of existing research.

We anticipate that a variety of augmentations, paired with a more detailed study of how different compositions of domain data influence representations, could bolster the expressivity of these topology-pre-trained models. Lastly and most significantly the work here is limited to labelled and un-labelled graphs, whereas most domains carry far more detailed features on nodes and edges. We view this as the most critical area for future research.

6 Conclusion

We have presented a graph representation model FOTOM trained through adversarial contrastive learning on a large, multi-domain dataset of unlabelled graphs. Under fine-tuning on diverse set of downstream tasks we demonstrate that this model trains faster, reaches better performance, and has more consistent performance than either non-pre-trained models or models pre-trained on domain specific datasets. This includes transfer where node labels are included in the fine-tuning data, which in itself is an important result worthy of further research. We envision that this model has broad use-cases as a foundational model, and that future works can both improve efficacy and extend our work to include node and edge features beyond labels.

References

[1] Maciej Besta, Torsten Hoefler, Maciej Besta, and Torsten Hoefler. 2018. Survey and Taxonomy of Lossless Graph Compression and Space-Efficient Graph Representations. *arXiv* (2018), arXiv:1806.01799. https://doi.org/10.48550/ARXIV.1806.01799

[2] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam Mccandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, Vol. 33. 1877–1901. https://commoncrawl.org/the-data/

[3] Hongyun Cai, Vincent W. Zheng, and Kevin Chen Chuan Chang. 2018. A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (9 2018), 1616–1637. https://doi.org/10.1109/TKDE.2018.2807452

[4] Fenxiao Chen, Yun Cheng Wang, Bin Wang, and C. C.Jay Kuo. 2020. Graph representation learning: A survey. , e15 pages. https://doi.org/10.1017/ATSIP.2020.13

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 1597–1607. https://proceedings.mlr.press/v119/chen20j.html

[6] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. 2021. CuCo: Graph Representation with Curriculum Contrastive Learning.. In *IJCAI*. 2300–2306.

[7] Alex Davies and Nirav Ajmeri. 2022. Realistic Synthetic Social Networks with Graph Neural Networks. (12 2022). http://arxiv.org/abs/2212.07843

[8] P. Erdos and A. Renyi. 1960. ON THE EVOLUTION OF RANDOM GRAPHS. (1960).

[9] Romanos Fasoulis, Georgios Paliouras, and Lydia E. Kavraki. 2021. Graph representation learning for structural proteomics. , 789–802 pages. https://doi.org/10.1042/ETLS20210225

[10] Hakim Hafidi, Mounir Ghogho, Philippe Ciblat, and Ananthram Swami. 2022. Negative sampling strategies for contrastive self-supervised learning of graph representations. *Signal Processing* 190 (2022), 108310.

[11] William L Hamilton. 2020. Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14, 3 (2020), 1–159.

[12] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. *arXiv* (9 2017). http://arxiv.org/abs/1709.05584

[13] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. 4116–4126.

[14] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*. International Conference on Learning Representations, ICLR.

[15] Licheng Jiao, Jie Chen, Fang Liu, Shuyuan Yang, Chao You, Xu Liu, Lingling Li, and Biao Hou. 2023. Graph Representation Learning Meets Computer Vision: A Survey. *IEEE Transactions on Artificial Intelligence* 4, 1 (2 2023), 2–22. https://doi.org/10.1109/TAI.2022.3194869

[16] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. 2021. Understanding Dimensional Collapse in Contrastive Self-supervised Learning. In *ICLR 2022 - 10th International Conference on Learning Representations*. International Conference on Learning Representations, ICLR. https://arxiv.org/abs/2110.09348v3

[17] Sammy Khalife, Khalife@lix Polytechnique Fr, Douglas Gonçalves, Youssef Allouah, Leo Liberti, and Liberti@lix Polytechnique Fr. 2021. Further results on latent discourse models and word embeddings. *The Journal of Machine Learning Research* 1 (1 2021), 1–36. https://doi.org/10.5555/3546258.3546528

[18] Shima Khoshraftar, Aijun An, Shima Khoshraftar, and Aijun An. 2022. A Survey on Graph Representation Learning Methods. *arXiv* (2022), arXiv:2204.01855. https://doi.org/10.48550/ARXIV.2204.01855

[19] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* 6, 1 (1 2009), 29–123. https://doi.org/10.1080/15427951.2009.10129177

[20] Haoyang Li, Xin Wang, Ziwei Zhang, Zehuan Yuan, Hang Li, and Wenwu Zhu. 2021. Disentangled contrastive learning on graphs. *Advances in Neural Information Processing Systems* 34 (2021), 21872–21884.

[21] Michelle M. Li, Kexin Huang, and Marinka Zitnik. 2022. Graph representation learning in biomedicine and healthcare. *Nature Biomedical Engineering* 6, 12 (12 2022), 1353–1369. https://doi.org/10.1038/s41551-022-00942-x

[22] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163. https://doi.org/10.1023/A:1009953814988/METRICS

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10684–10695. https://github.com/CompVis/latent-diffusion

[24] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-Scale attributed node embedding. *Journal of Complex Networks* 9, 2 (5 2021), 1–22. https://doi.org/10.1093/comnet/cnab014

[25] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *CIKM '20: Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. ACM, Online, Ireland, 3125–3132. http://arxiv.org/abs/2003.04819

[26] Pascal Schweitzer PASCAL, Erik Jan van Leeuwen EJVANLEEUWEN, Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, Karsten M Borgwardt SHERVASHIDZE, and Van Leeuwen. 2011. Weisfeiler-Lehman Graph Kernels Nino Shervashidze Kurt Mehlhorn Karsten M. Borgwardt. In *Journal of Machine Learning Research*, Vol. 12. 2539–2561.

[27] Hamed Shirzad, Kaveh Hassani, and Danica J. Sutherland. 2022. Evaluating Graph Generative Models with Contrastively Learned Features. In *Advances in Neural Information Processing Systems*, Vol. 35. 7783–7795. https://github.com/hamed1375/

[28] Arjun Subramonian. 2021. Motif-driven contrastive learning of graph representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 15980–15981.

[29] Susheel Suresh, Pan Li, Cong Hao, Georgia Tech, and Jennifer Neville. 2021. Adversarial Graph Augmentation to Improve Graph Contrastive Learning. In *Advances in Neural Information Processing Systems*, Vol. 34. 15920–15933. https://github.com/susheels/adgcl

[30] Da CostaVictor G. Turrisi, FiniEnrico, NabiMoin, SebeNicu, and RicciElisa. 2022. solo-learn: a library of self-supervised methods for visual representation learning. *The Journal of Machine Learning Research* 23 (1 2022), 1–6. https://doi.org/10.5555/3586589.3586645

[31] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax.

[32] Michael Winding, Benjamin D. Pedigo, Christopher L. Barnes, Heather G. Patsolic, Youngser Park, Tom Kazimiers, Akira Fushiki, Ingrid V. Andrade, Avinash Khandelwal, Javier Valdes-Aleman, Feng Li, Nadine Randel, Elizabeth Barsotti, Ana Correia, Richard D. Fetter, Volker Hartenstein, Carey E. Priebe, Joshua T. Vogelstein, Albert Cardona, and Marta Zlatic. 2023. The connectome of an insect brain. *Science* 379, 6636 (3 2023). `https://doi.org/10.1126/science.add9330`

[33] Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2017. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science* 9, 2 (3 2017), 513–530. `https://doi.org/10.1039/c7sc02664a`

[34] Dongkuan Xu, Wei Cheng, Dongsheng Luo, Haifeng Chen, and Xiang Zhang. 2021. Infogcl: Information-aware graph contrastive learning. *Advances in Neural Information Processing Systems* 34 (2021), 30414–30425.

[35] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[36] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. 2020. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533* (2020).

[37] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An empirical study of graph contrastive learning. *arXiv preprint arXiv:2109.01116* (2021).

[38] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*. 2069–2080.

A Supplemental Material

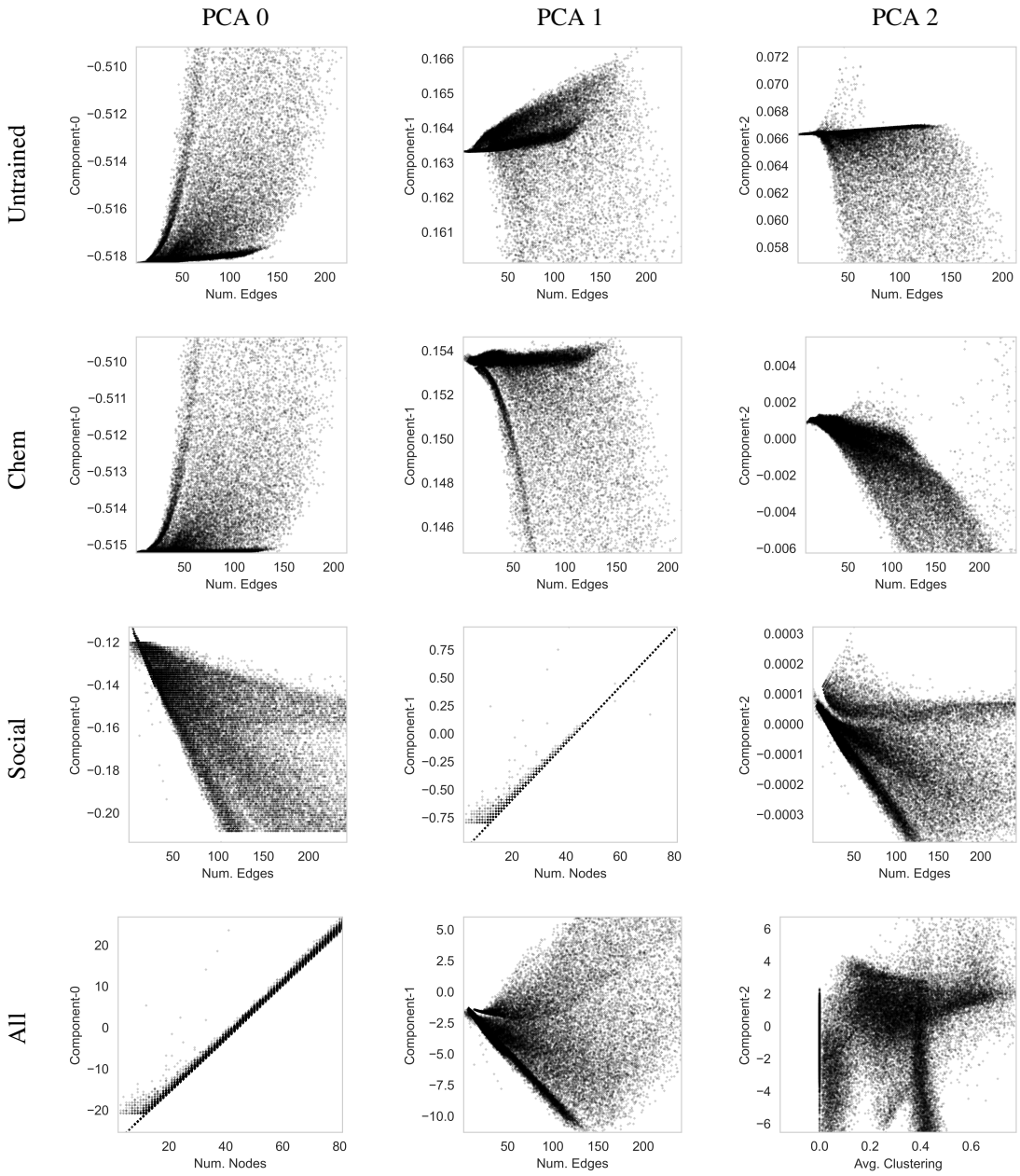


Figure 3: PCA components of the encodings of each model across the whole validation dataset, scattered against the most correlated metrics, as in Table 2.

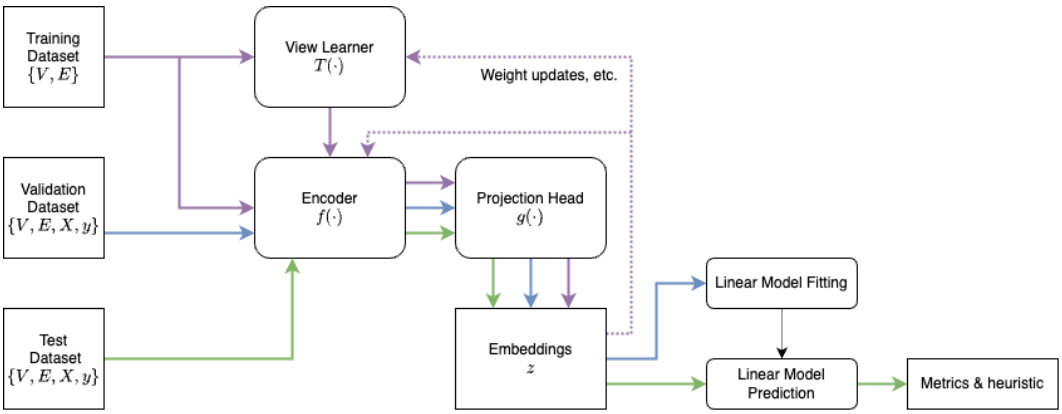


Figure 4: Our evaluation process. V represents nodes, E edges, X node labels (where available and if included) and y graph target values.

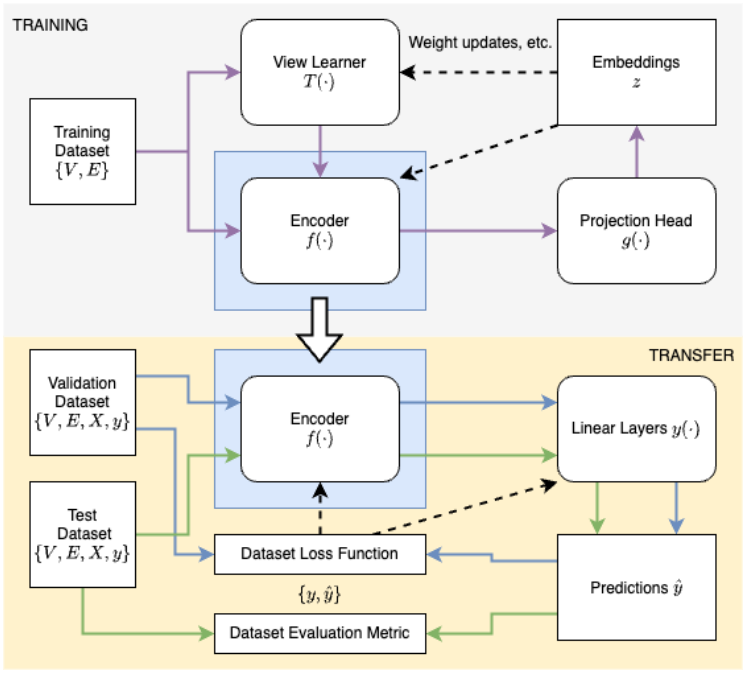


Figure 5: Our evaluation process for transfer learning. V represents nodes, E edges, X node labels (where available and if included) and y graph target values.

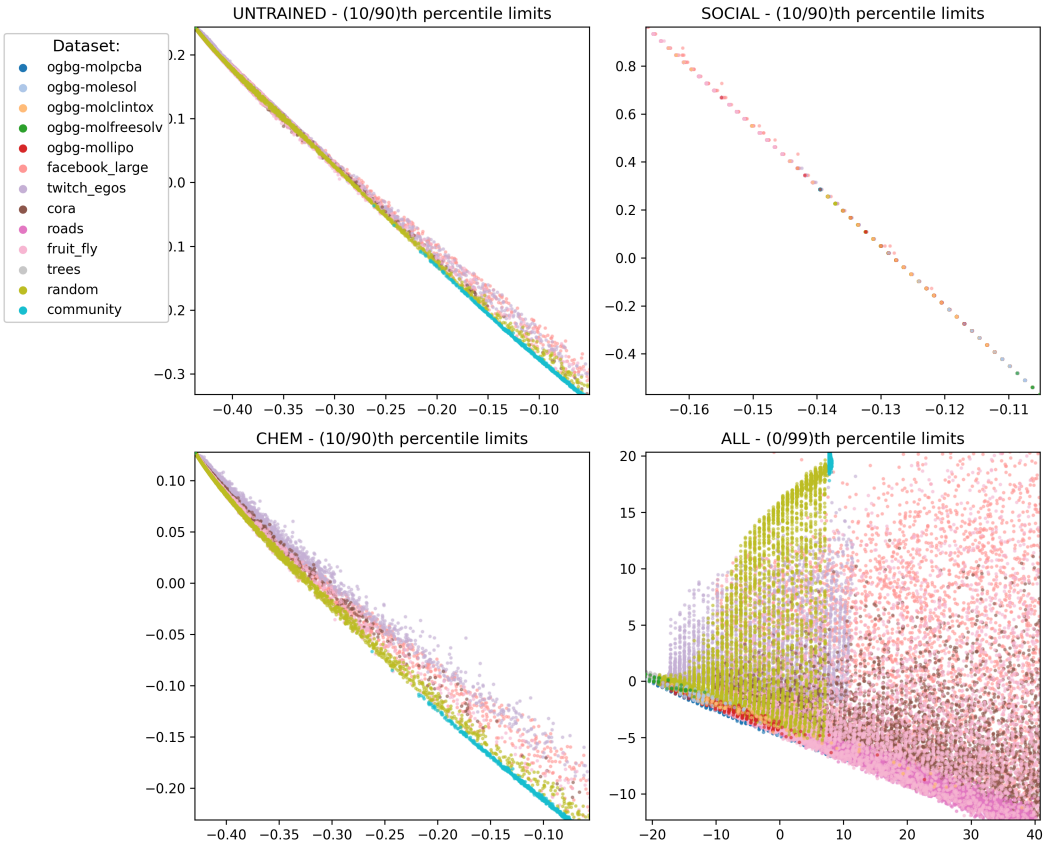
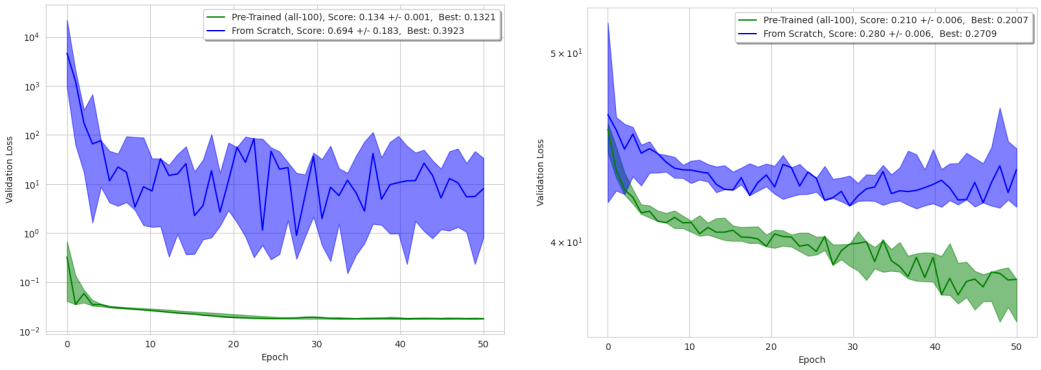


Figure 6: PCA projections of encodings from each model, as well as an untrained model. We scale axes by percentiles - of - data - included as for all except the All model the projections have outliers in the region 10^7.



(a) Facebook

(b) Twitch Egos

Figure 7: Validation scores for fine-tuning on the Facebook and Twitch datasets. On the Facebook dataset the task is average clustering prediction, and Twitch dataset carries its binary classification task. We include the average scores and the best scores over the ten training runs on each dataset.

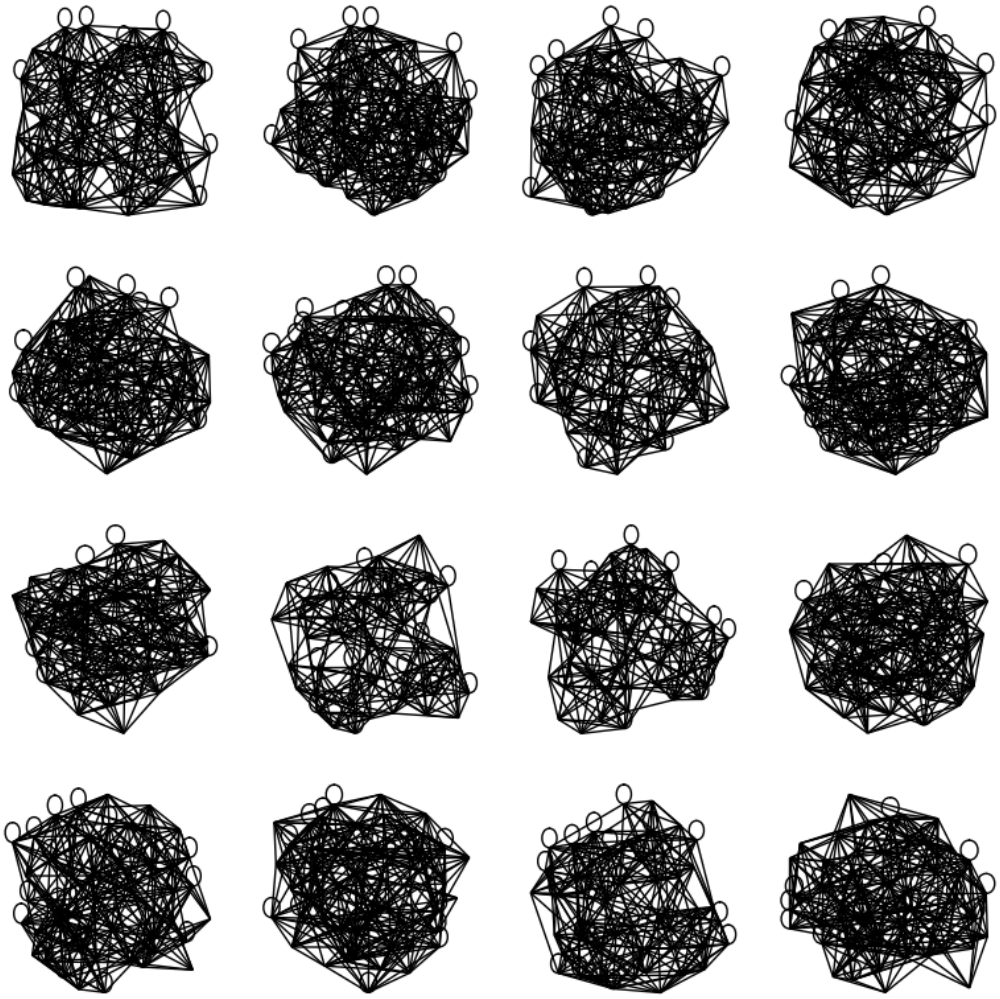


Figure 8: Graphs from the community dataset

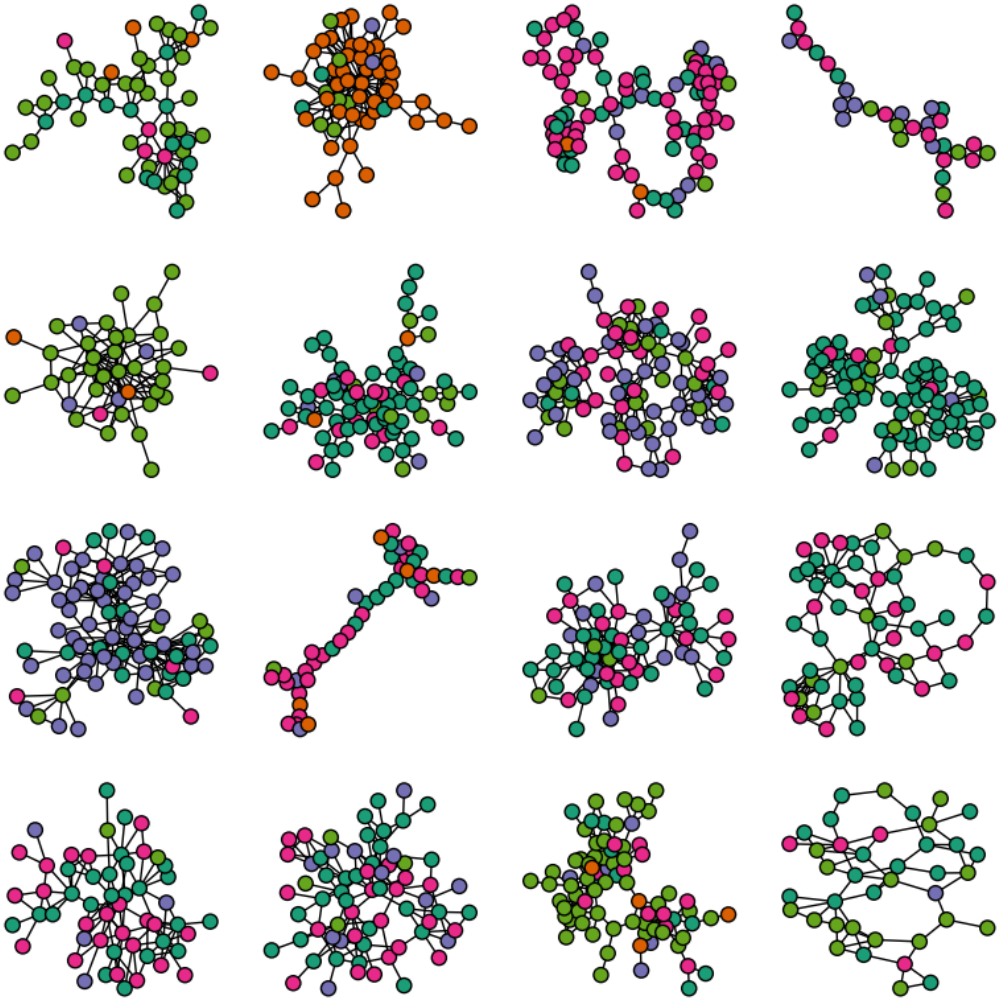


Figure 9: Graphs from the CORA dataset [22]

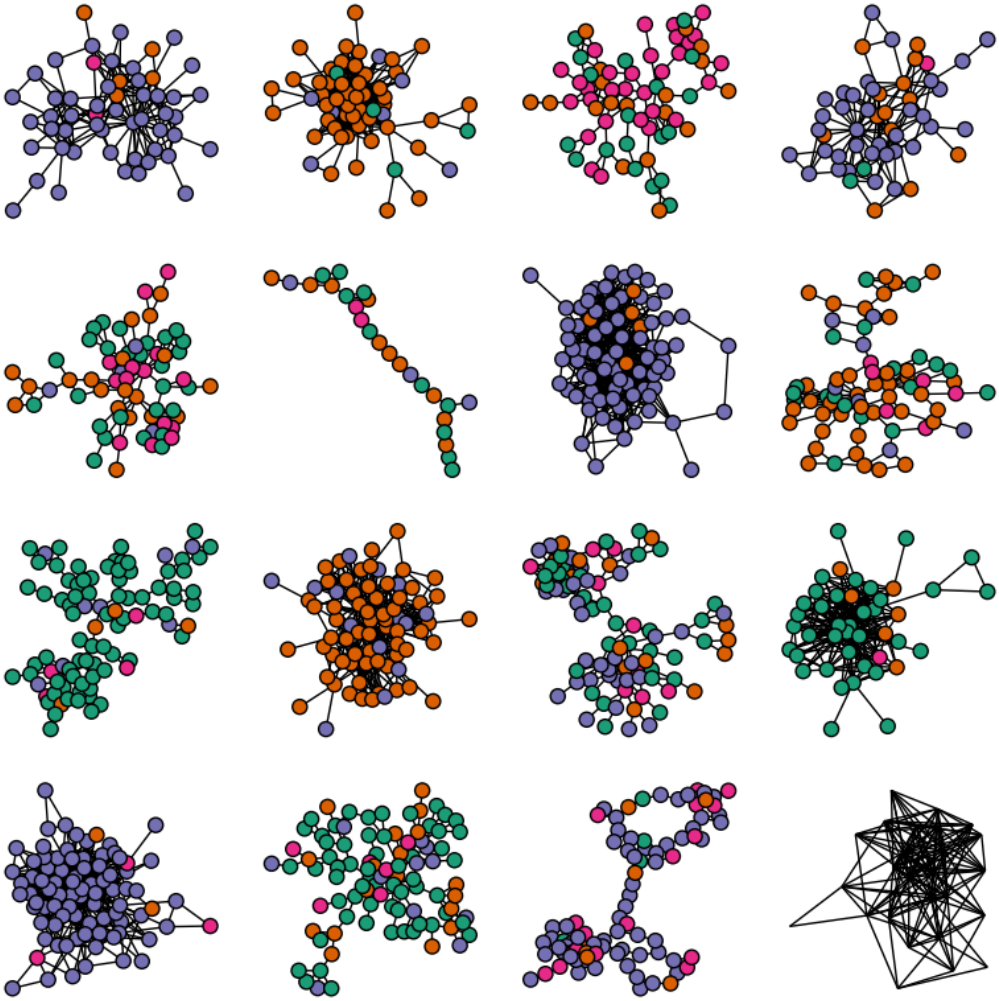


Figure 10: Graphs from the Facebook dataset [24]

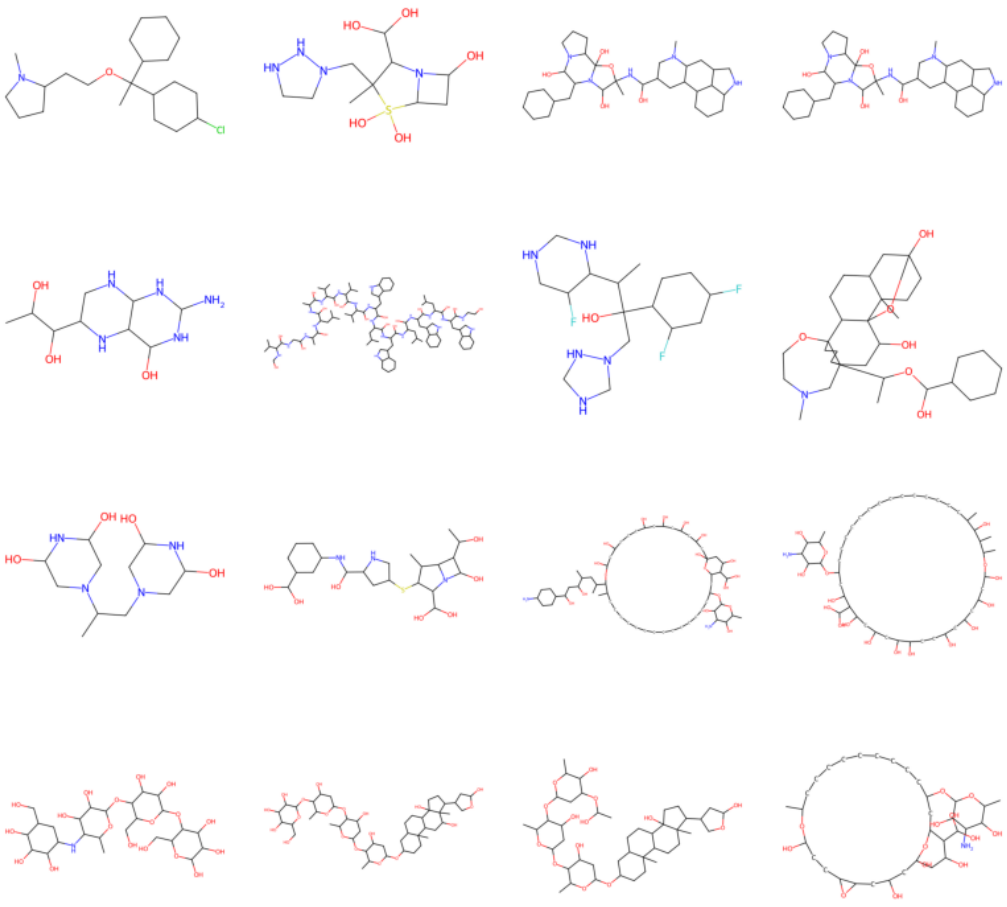


Figure 11: Graphs from the ogbg-molclintox dataset [14]

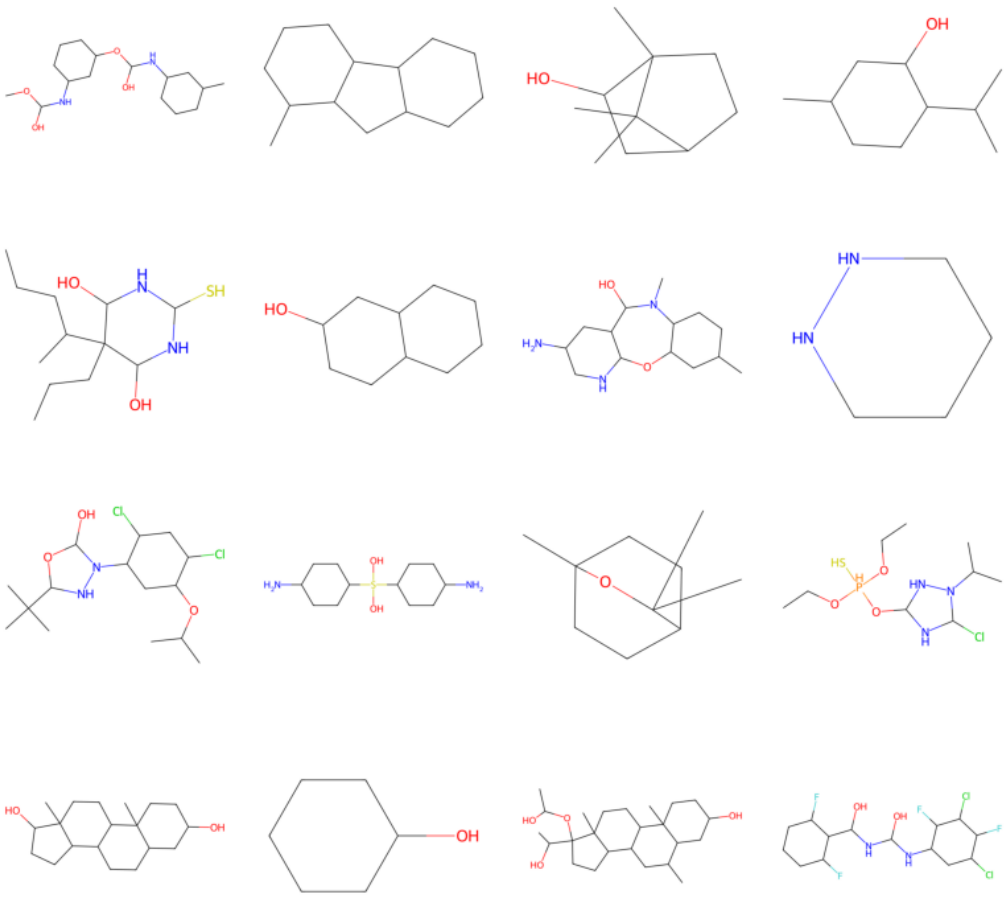


Figure 12: Graphs from the ogbg-molesol dataset [14]

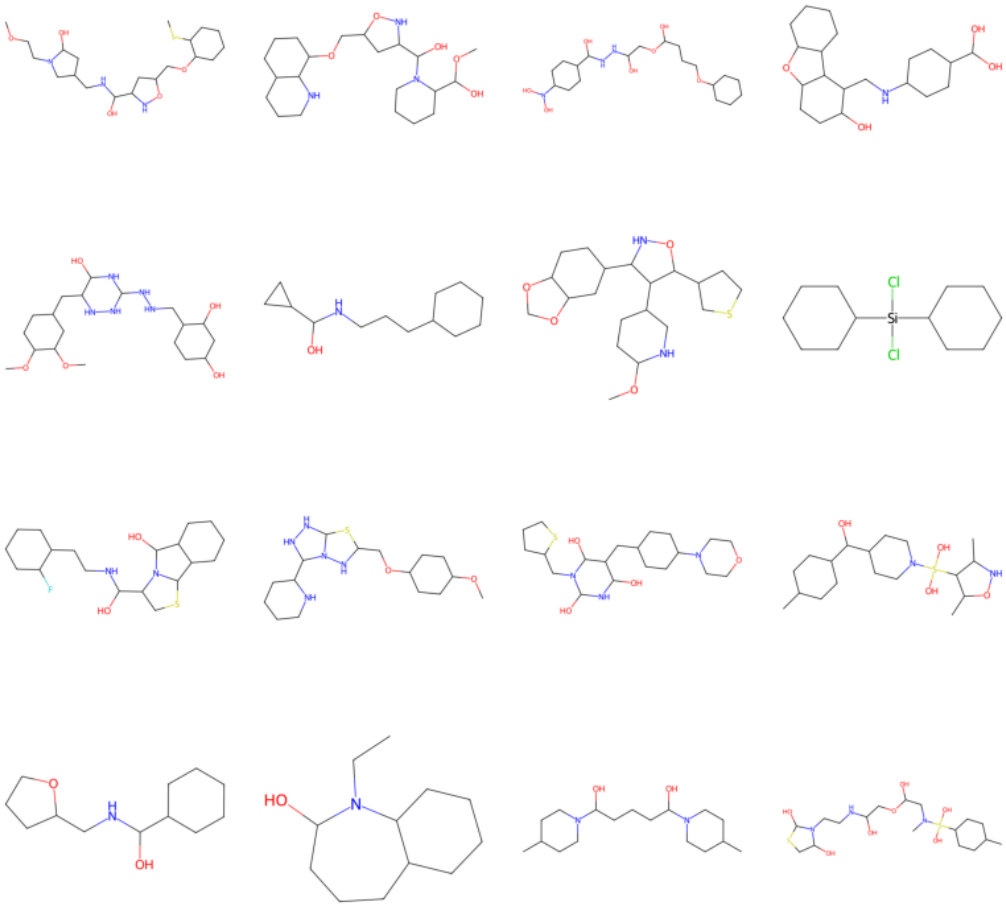


Figure 13: Graphs from the ogbg-molpcba dataset [14]

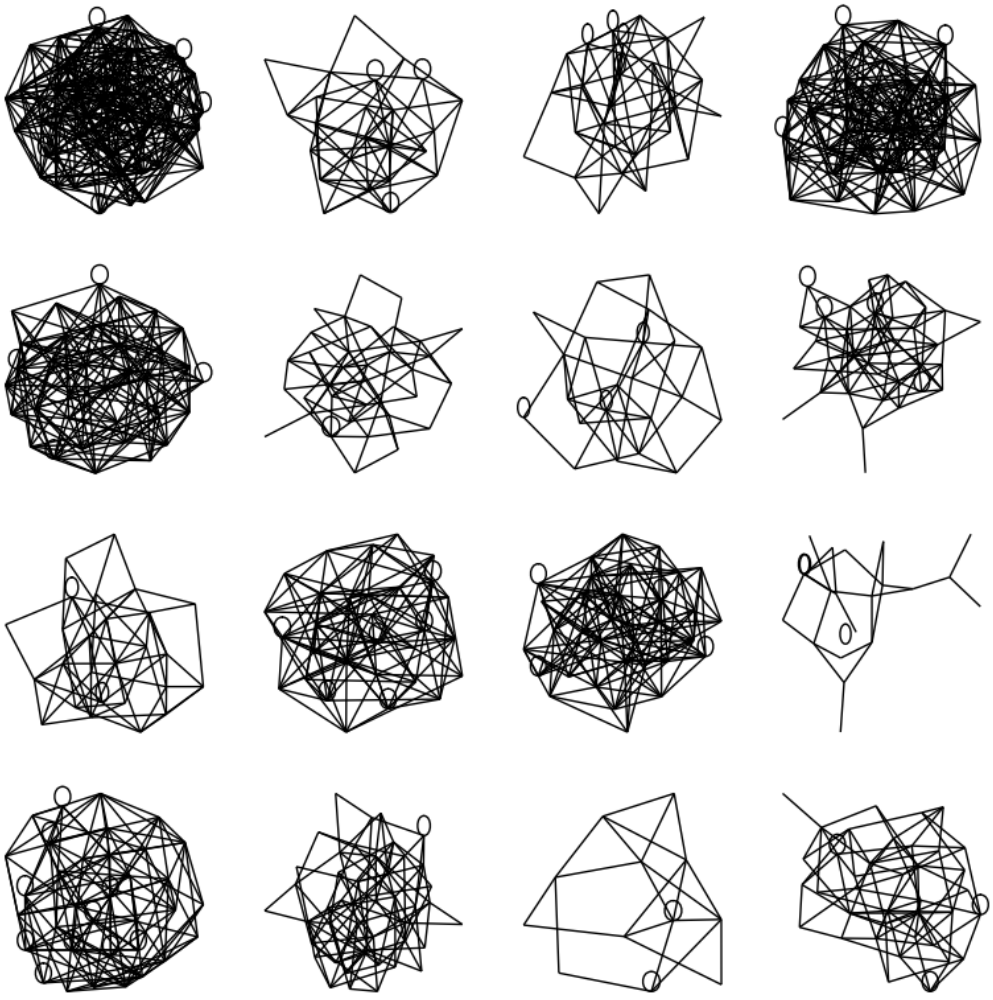


Figure 14: Graphs from the random dataset

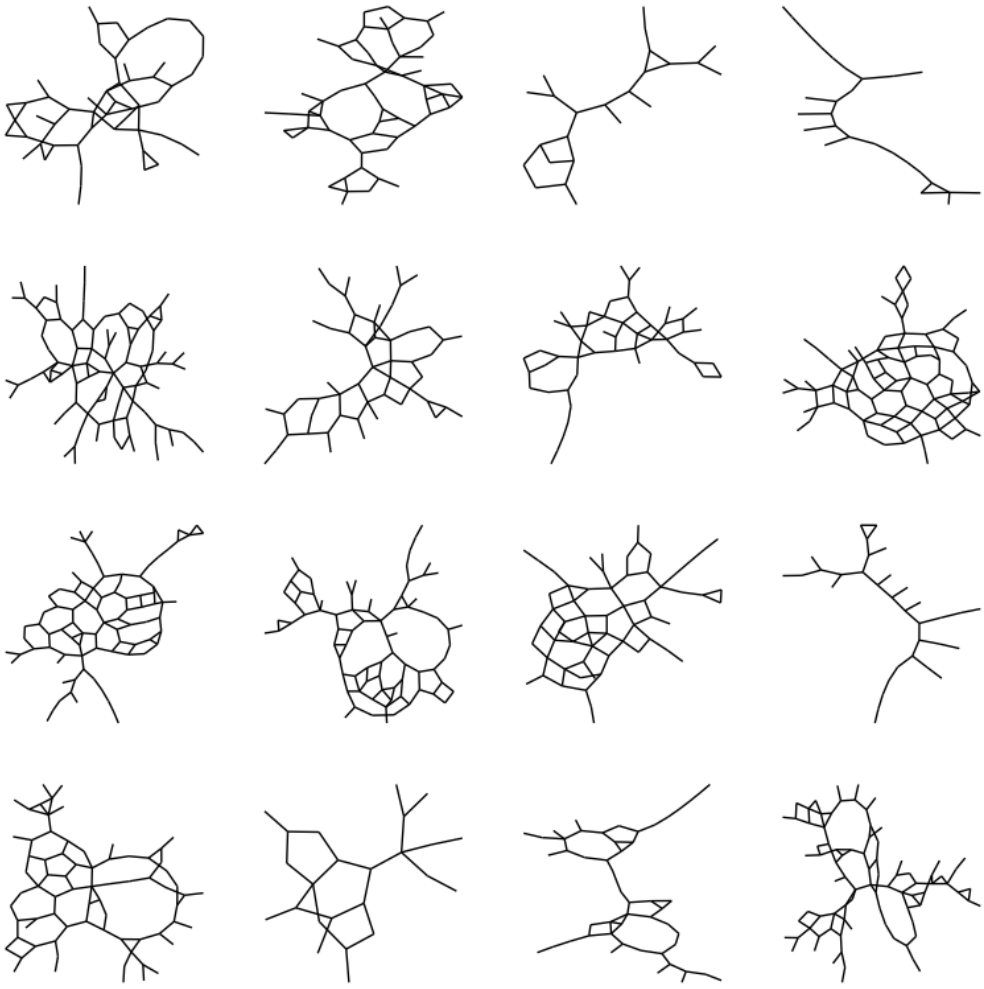


Figure 15: Graphs from the roads dataset [19]

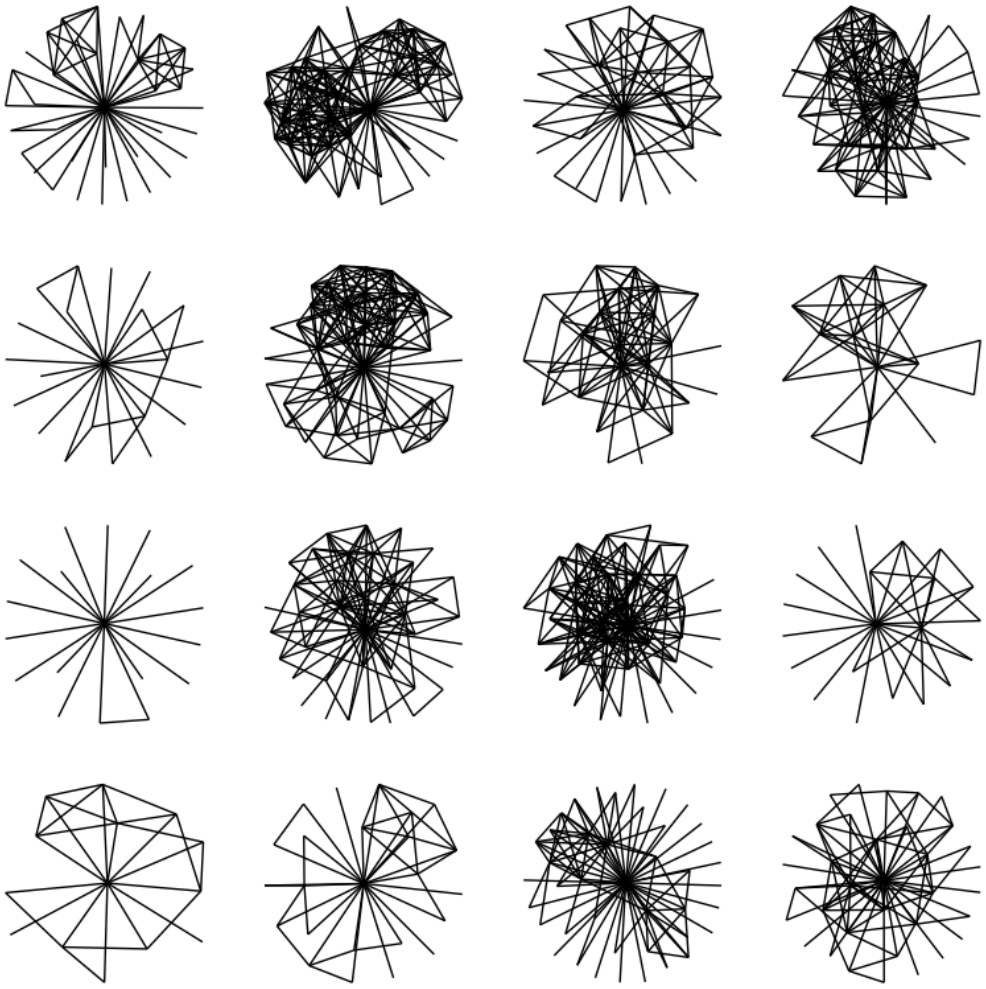


Figure 16: Graphs from the twitch egos dataset [25]

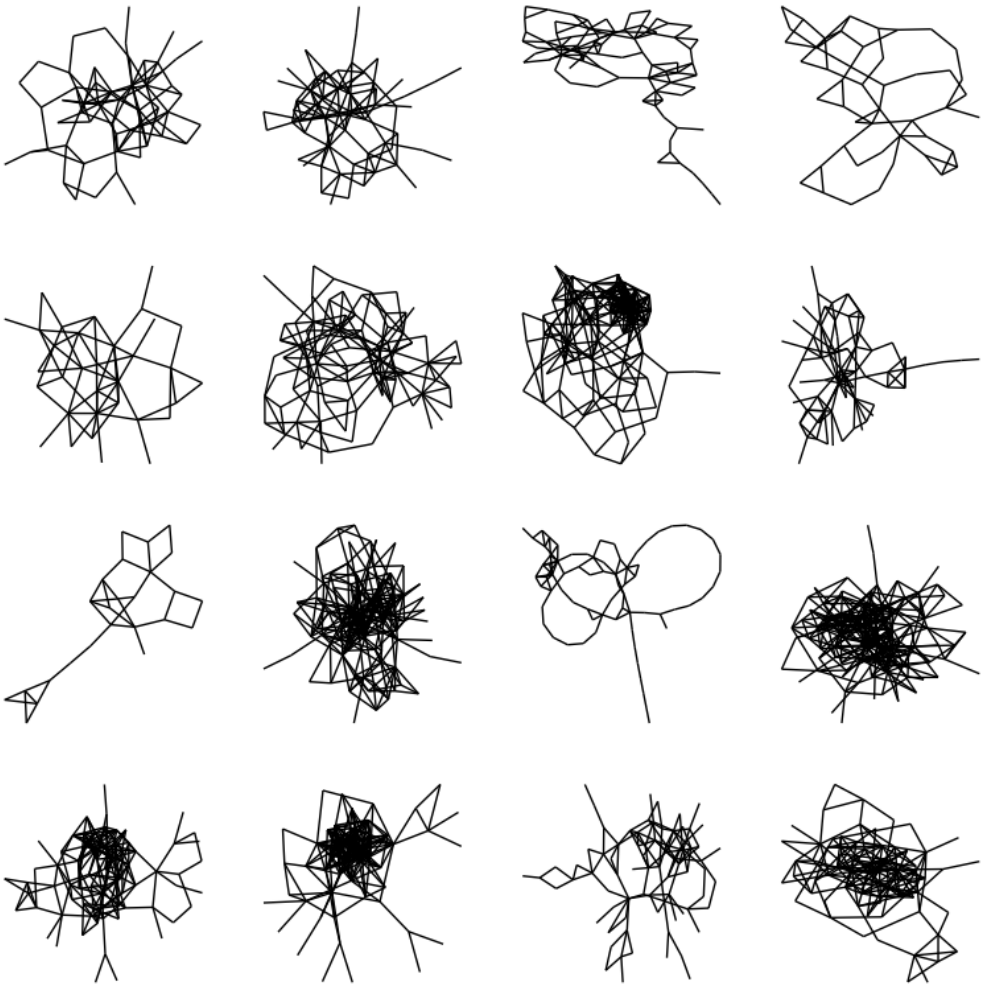


Figure 17: Graphs from the fruit fly brain dataset [32]

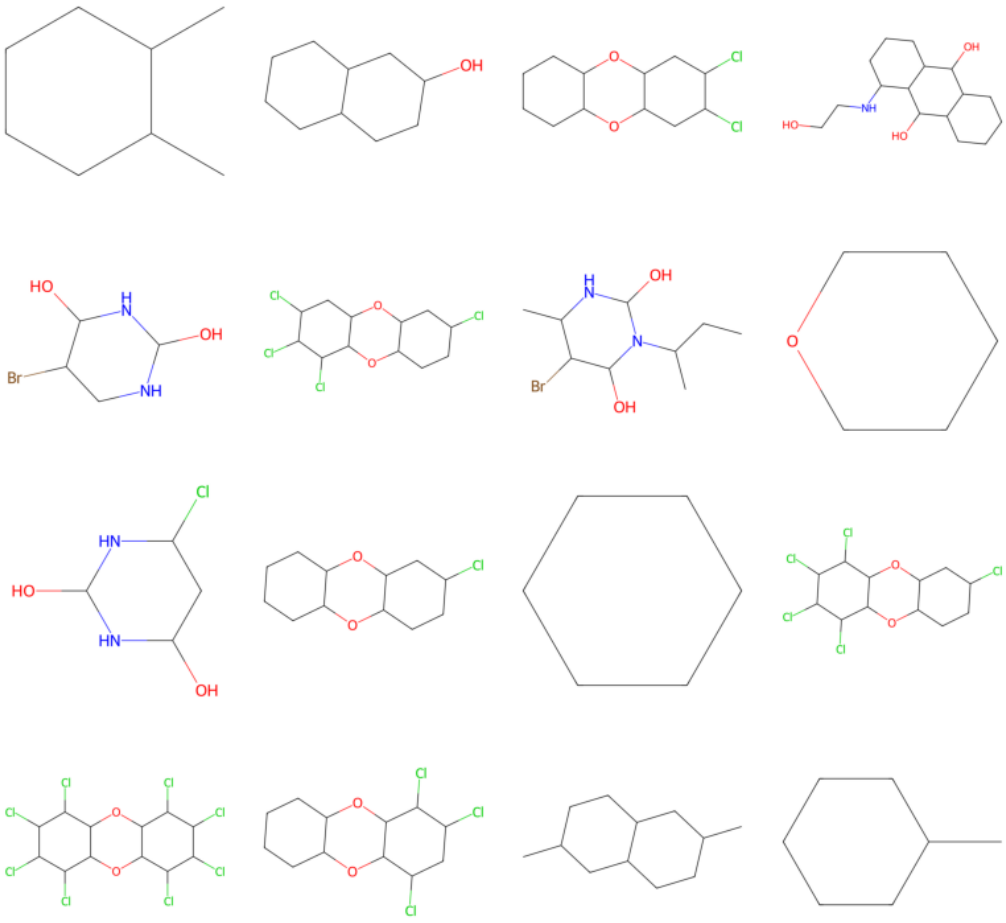


Figure 18: Graphs from the molfreesolv dataset [14]

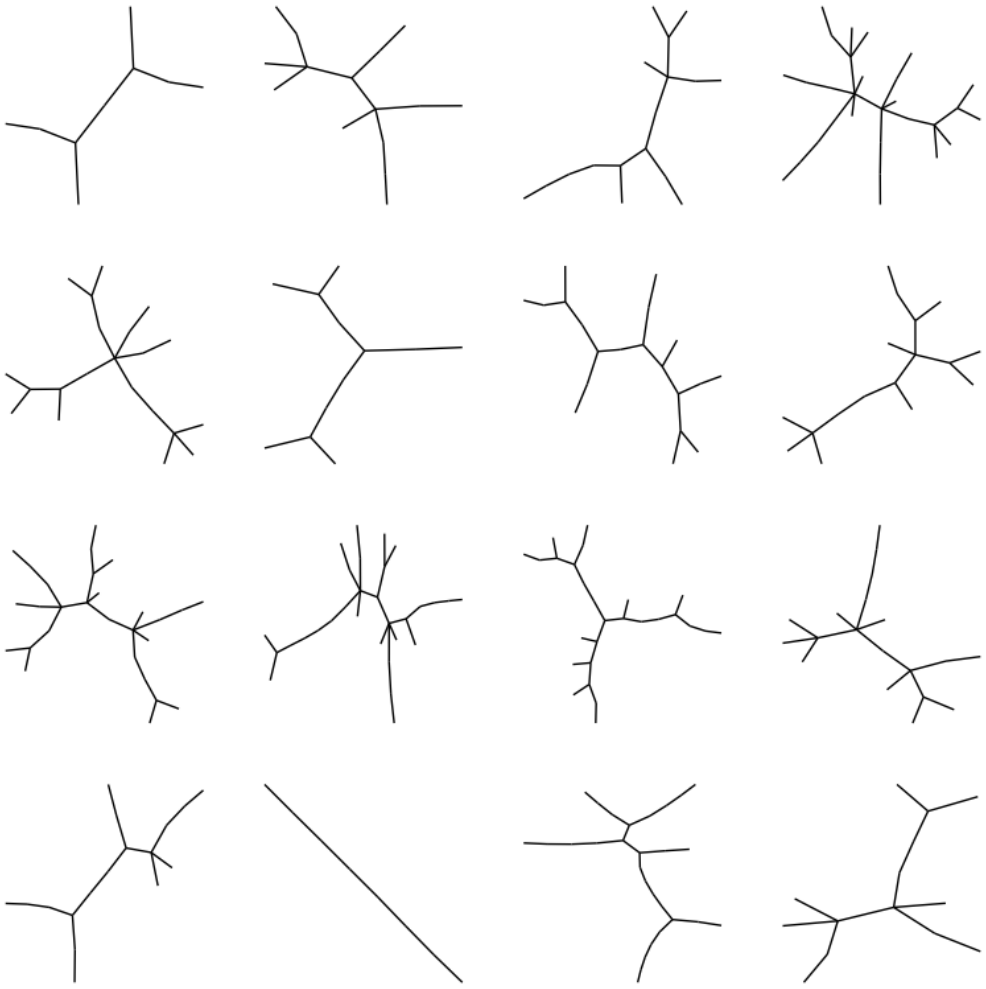


Figure 19: Graphs from the trees dataset

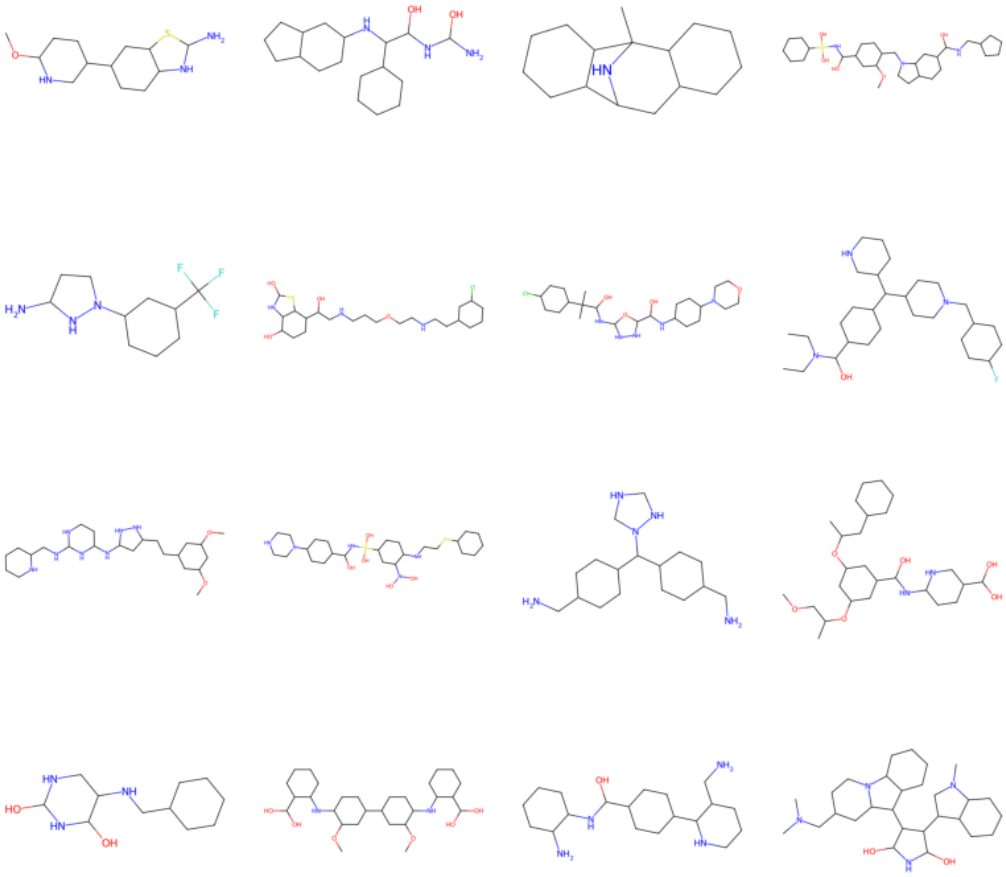


Figure 20: Graphs from the mollipo dataset [14]